General Informatics - Main Contents

Chapter 2. BOOLEAN Algebra

2.1. Introduction

George BOOLE (1815-1864), the famous English mathematician, published in 1854 the book "An Investigation of the Law of Thought", in which he put the bases of Boolean algebra.

Later on, this new theory of mathematics has been developed by E. SCHRÖDER, A.N. WITEHEAD, B. RUSSEL and C.E. SHANNON. This last, in 1938, in his work "Analysis of Relay and Switching Circuit" introduces for the first time the naming of "**AND GATE**" and "**OR GATE**".

A special contribution in the development of Boolean algebra had Romanian school, leaded by Gr. MOISIL.

Introduction

- In digital computers are used digital electronic devices which may have only two voltage level referred to as logic "1" and logic "0" states and, as "True" and "False".
- Because of the use of only two states, digital logic is said to be **binary** in nature. Thus, logic circuits can carry out all of their decision-making and memory functions by using no more than the two states.
- In Boolean algebra, a sentence can be true or false, but never true and false in the same time. Two sentences are equivalent if they are both true or false by once.
- Let, for example, the sentence "transistor T is spend". Instead of writing this sentence, it's easier to write a *variable* T, which can be: T = 1 when the sentence is true; T = 0 when the sentence is false.
- By introducing the BOOL multitude $\mathbf{B}_2 = \{0,1\}$ we can say that the variable $T \in \mathbf{B}_2$.
- Some common representation of 0 and 1:

  Logic 0 -> False, Off, Low, No, Open Switch

  Logic 1 -> True, On, High, Yes, Close switch

Definition

- *Definition*: A Boolean algebra is a multitude of elements B_2, with two laws of composition noted with "+" (or noted "$\cup$") and "." (or noted "$\cap$") named Boolean sum and Boolean product and a low of complementation denoted by "$\bar{\ }$" (not).
- The symbols "+" and "." are called *logical connectives;* they should not be confused with + sign and . decimal point of conventional algebra.
- The logical functions, in Boolean algebra, can be represented by *logical equations* or by *truth table.*
- The *truth table* is a means for describing how a logic circuit's output depends on the logic levels present at the circuit's input.

Alternative Definitions:

- *Webster Dictionary: an algebraic system that consists of a set of closed under two binary operations and that can be described by any of various systems of postulates all of which can be deduced from the postulates that each operation is commutative, that each operation is distributive over the other, that an identity element exists for each operation, and that for every element in the 1st there exists another element which when combined to the 1st under either one of operations yields the identity element of the other operation.*

- *MSN Encarta: algebra concerned with binary combinations: a form of algebra concerned with the logical functions of variables that are restricted to two values, true or false. Boolean algebra is fundamental to circuit design and to the design, function and operation of computers.*

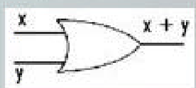
Rules in Boolean Algebra

The syntactical rules in Boolean algebra are:

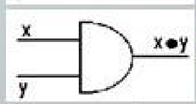
$$not(x) \Rightarrow \bar{x}$$
$$and(x, y) \Rightarrow x \bullet y$$
$$or(x, y) \Rightarrow x + y$$

Given x, y$\in B_2$ the logic equations are defined as:

OR function: $f(x,y) = x \cup y = x + y$



AND function: $f(x,y) = x \cap y = x \cdot y$



NOT function: $f(x) = \bar{x}$

Fundamental Theorems of Boolean Algebra

1. *Law of Oneness:*
   - The element 0 is unique. It exists an element 0, called first element so that:

     $x \cdot 0 = 0$

     $x + 0 = x$
   - The element 1 is unique. It exists an element 1, called last element so that:

     $x \cdot 1 = x$

     $x + 1 = 1$
2. *Law of Complementation:*

   $x \bullet \bar{x} = 0$

   $x + \bar{x} = 1$
3. *Law of Double Complementation*

   $\bar{\bar{x}} = x$
4. *De Morgan's Theorem*

   $\overline{x \bullet y} = \bar{x} + \bar{y}$

   $\overline{x + y} = \bar{x} \bullet \bar{y}$
5. *Law of Absorption*

   $x + (x \bullet y) = x$

   $x \bullet (x + y) = x$
6. *Idempotent Rule* (or Law of Tautology)

   $x + x = x$

   $x \bullet x = x$
7. *Laws of Reunion and Intersection*

   $x + 0 = x$

   $x + 1 = 1$

   $x \bullet 1 = x$

   $x \bullet 0 = 0$
8. *Law of Commutation*

   $x \bullet y = y \bullet x$

   $x + y = y + x$
9. *Law of Association*

   $x + y + z = (x + y) + z = x + (y + z)$

   $(x \bullet y) \bullet z = x \bullet (y \bullet z)$
10. *Law of Distribution*

    $x \bullet (y + z) = x \bullet y + x \bullet z$

    $x + y \bullet z = (x + y) \bullet (x + z)$

Fundamental Theorems of Boolean Algebra

- *Duality Principle: the dual is obtained by interchanging 'AND' and 'OR' operators and by replacing 0's by 1's and 1's by 0's.*

Example *(consensus theorem)*:

$$x \bullet y + \bar{x} \bullet z + y \bullet z = x \bullet y + \bar{x} \bullet z$$

or the dual form:

$$(x + y) \bullet (\bar{x} + z) \bullet (y + z) = (x + y) \bullet (\bar{x} + z)$$

Proof of: $x \bullet y + \bar{x} \bullet z + y \bullet z = x \bullet y + \bar{x} \bullet z$

$$x \bullet y + \bar{x} \bullet z + (x + \bar{x}) \bullet y \bullet z = x \bullet y + \bar{x} \bullet z + x \bullet y \bullet z + \bar{x} \bullet y \bullet z =$$

$$x \bullet y \bullet (1 + z) + \bar{x} \bullet z \bullet (1 + y) = x \bullet y + \bar{x} \bullet z$$

Fundamental Theorems of Boolean Algebra

- *Simplification theorems*

$$x \bullet y + x \bullet \overline{y} = x \quad (uniting) \quad (x+y) \bullet (x+\overline{y}) = x$$

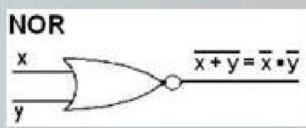
$$x + x \bullet y = x \quad (absorption) \quad x \bullet (x+y) = x$$

$$(x+\overline{y}) \bullet y = x \bullet y \quad (absorption) \quad x \bullet \overline{y} + y = x + y$$

Fundamental Theorems of Boolean Algebra

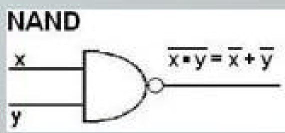
NOR

x	y	$x+y$	$\overline{x+y}$	$\overline{x} \bullet \overline{y}$	$\overline{x}$	$\overline{y}$
0	0	0	1	1	1	1
0	1	1	0	0	1	0
1	0	1	0	0	0	1
1	1	1	0	0	0	0



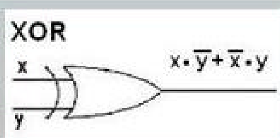
NAND

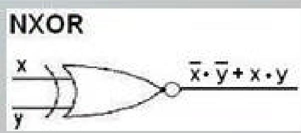
x	y	$\overline{x}$	$\overline{y}$	x•y	$\overline{x \bullet y}$	$\overline{x} + \overline{y}$
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0



Fundamental Theorems of Boolean Algebra

x	y	XOR (exclusive OR)	NOT XOR (NXOR)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1





The Existence and Oneness of Boolean Functions

I. Boolean function of one variable

$f : B_2 \to B_2$ and a, b constants, $a, b \in B_2 = \{0,1\}$

NDF – Normal Disjunctive Form $f_1(x) = a \bullet x + b \bullet \overline{x}$

NCF – Normal Conjunctive Form $f_2(x) = (a + x) \bullet (b + \overline{x})$

II. Boolean function of two variables

$f : B_2 x B_2 \to B_2$ and a, b, c, d constants, $a, b, c, d \in B_2 = \{0,1\}$

NDF – Normal Disjunctive Form $f_1(x, y) = a \bullet x \bullet y + b \bullet \overline{x} \bullet y + c \bullet x \bullet \overline{y} + d \bullet x \bullet y$

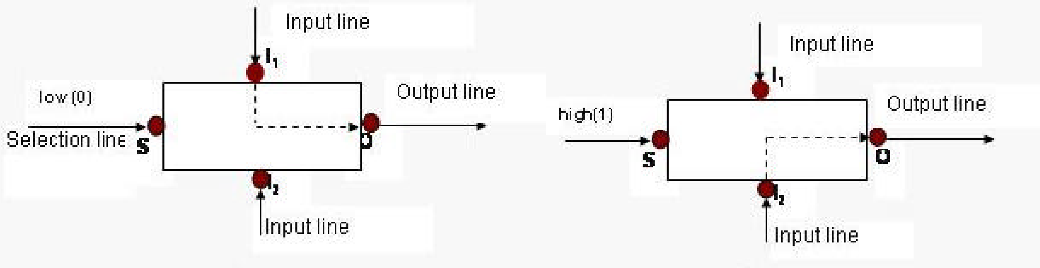
NCF – Normal Conjunctive Form $f_2(x, y) = (a + x + y) \bullet (b + \overline{x} + y) \bullet (c + x + \overline{y}) \bullet (d + \overline{x} + \overline{y})$

Binary Switches

The functionality is based on the value of the selection signal used to select only one input line. The behavior can be defined as:

The value at select line is low ⇒ select input line I1 (figure 2.2 a)

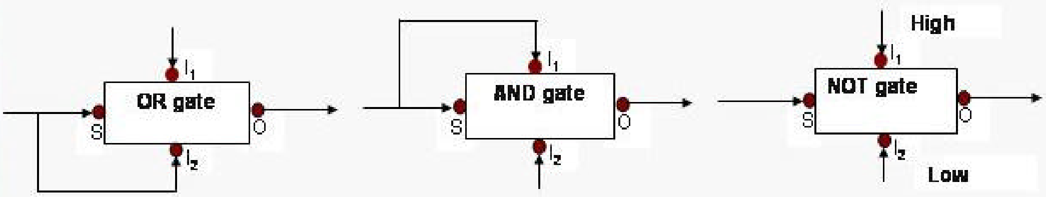
The value at select line is high ⇒ select input line I2 (figure 2.2 b)



a) b)

Figure 2. 2 Binary switches –external view

Binary Switches



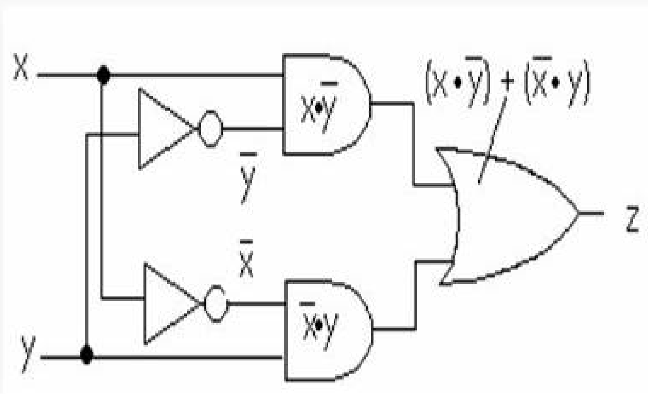


Figure 2. 8 The implementation of xor(x,y) function

Implementing Logical Functions

$$\begin{cases} z = (\overline{x} \cdot y) + (x \cdot \overline{y}) \\ c_{out} = x \cdot y \end{cases}$$

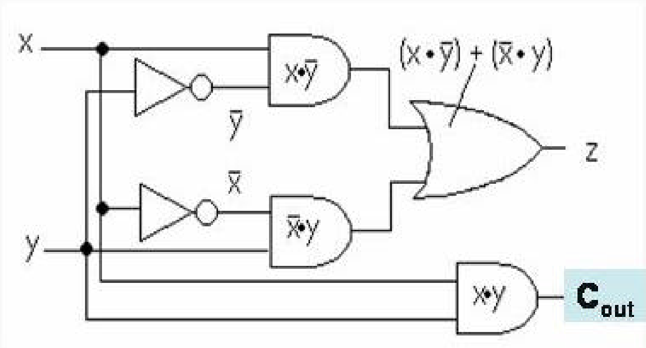


Figure 2. 9 The implementation of the simple calculator (a binary half adder)

Implementing Logical Functions

There are two basic forms for a Boolean function (canonical forms):

- **sum-of-products statement** (or *minterm* form or disjunctive canonical form - DCF) in which variables or their complements are connected by AND, and *minterms* are connected by OR;
- **product-of-sums statement** (or *maxterm* form or conjunctive canonical form - CCF), in which variables or their complements are connected by OR and *maxterms* are connected by AND.

For example, let's consider:

$$f(x, y, z): B_2 \times B_2 \times B_2 \to B_2$$

then the CCF and DCF forms of the function are defined as:

$$f(x,y,z) = (x+y+z) \cdot (x+\overline{y}+z) \cdot (x+y+\overline{z}) \quad \text{(CCF)}$$

$$f(x,y,z) = x \cdot y \cdot z + \overline{x} \cdot y \cdot z + x \cdot \overline{y} \cdot z + x \cdot y \cdot \overline{z} \quad \text{(DCF)}$$

Implementing Logical Functions

The names of Maxterm and minterm are explained by the use of Venn diagram:

▪$(x \cdot y \cdot z)$ is called minterm because the hachured area in the diagram represents the value of $(x \cdot y \cdot z)$ and it is minimal (figure 2.13);

▪$(x + y + z)$ is called Maxterm because the hachured area is maximal (figure 2.14).

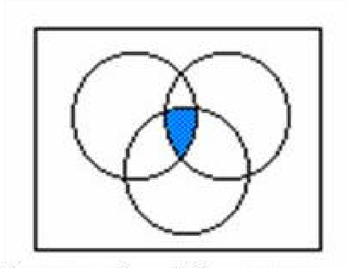


Figure 2. 13 x•y•z - minterm

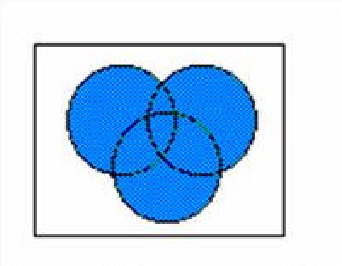


Figure 2. 14 x+y+z - Maxterm

The Main Properties of min and max terms

P1) $m_i \cdot m_j = 0 \; \forall \; i \neq j$

P2) $M_i + M_j = 1 \; \forall \; i \neq j$

P3) Any Boolean function can be expressed by a logical sum of minterms m_i or by a logical product of maxterms M_i

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=0}^{2^n - 1} \alpha_i \cdot m_i \qquad \text{(DCF)}$$

$$f(x_1, x_2, \ldots, x_n) = \prod_{i=0}^{2^n - 1} (\alpha_i + M_i) \qquad \text{(CCF)}$$

where $\alpha_i \in B2 = \{0,1\}$ and are named characteristic numbers.

P_4) If

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=0}^{2^n-1} \alpha_i \cdot m_i$$

then (DCF)

$$\overline{f}(x_1, x_2, \ldots, x_n) = \sum_{i=0}^{2^n-1} \overline{\alpha_i} \cdot m_i$$

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=0}^{2^n-1} (\alpha_i \cdot M_i)$$

then (CCF)

$$\overline{f}(x_1, x_2, \ldots, x_n) = \prod_{i=0}^{2^n-1} (\overline{\alpha_i} + M_i)$$

P_5) If a Boolean function of n variables is written in DCF and it contains 2^n distinct terms of n variables, then the value of this function is 1:

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=0}^{2^n-1} m_i = 1$$

For the same function, in the same conditions, if it is written in CCF, then it equals 0:

$$f(x_1, \ldots, x_n) = \prod_{i=0}^{2^n-1} M_i = 0$$

P_6) Any minterm m_i **of a Boolean function of n variables, written in DCF, equals a logical product of (2^n-1) terms** M_j:

$$m_i = \prod_{i \neq j} M_j$$

with j=0, 1, ..., 2^n-1

Respectively, any maxterm M_i of a Boolean function written in CCF equals a logical sum of (2^n-1) terms m_j:

$$M_i = \sum_{i \neq j} m_j$$

with j=0, 1, ..., 2^n-1

Regarding these properties of a Boolean function, we can observe that there are 2^{2^n} distinct functions for n binary variables.

For a function $f(x_1,...,x_n)$ there are 2^n minterms m_i and the logical sum of a term of several terms corresponds to a distinct Boolean function.

BOOLEAN ALGEBRA - General Contents

Bibliography

1.[Av.00]Avram Vasile - *Sisteme de calcul şi operare*, volumul I, Editura Dacia Europa Nova Lugoj, 2002

2. [AvDg.97]Avram Vasile, Dodescu Gheorghe - *General Informatics*, Editura Economică, Bucureşti, 1997 (you can find other simplification techniques for Boolean functions)

3. [AvDg.03]Avram Vasile, Dodescu Gheorghe – *Informatics: Computer Hardware and Programming in Visual Basic*, Editura Economică, Bucureşti, 2003, chapter 2, pages 53-82